

Application No. 09/751,762
Response dated October 2, 2006
Final Office Action dated May 30, 2006

Atty. Docket No. 2207/10121

REMARKS/ARGUMENT

Claims 1-21 are pending in the application. Claims 1-21 are rejected under 35 U.S.C. 102(e) as being anticipated by Eikemeyer (U.S. Patent No. 6,694,425) ("Eikemeyer").

Applicants respectfully submit that the cited references do not teach, suggest or disclose "[a] method comprising: ... flushing an instruction from said first thread from a pipeline of said processing system if data is to be loaded from said memory device before executing said instruction" (e.g., as described in claim 1).

The Office Action asserts that Eikemeyer teaches flushing an instruction from the thread of a pipeline of said processing system when data is to be loaded from [a] memory device before executing the instruction. See Office Action dated 12/21/05, page 4, paragraph 6. Applicants respectfully disagree. The Office Action cites column 1, lines 6-12; column 6, lines 18-26 and column 11, lines 41-49. Column 1, lines 6-12 state:

The present invention relates in general to an improved data processing system and in particular to an improved system and method for switching threads of execution when execution of a thread is stalled in the dispatch stage of a multithread pipelined processor and flushing the stalled thread from earlier stages of pipeline.

Applicants agree with the Office Action's assertion that this section includes the generic description of an instruction of a thread being flushed. See paragraph 6, line 10. The cited section describes an improved system for switching threads when execution of a thread is stalled in a multithreaded pipelined processor and flushing the stalled thread from earlier stages of a pipeline.

However, this generic disclosure of the purpose behind the Eikemeyer reference does not disclose "... flushing an instruction from said first thread from a pipeline of said processing system if data is to be loaded from [a] memory device before executing said instruction" (e.g., as described in claim 1). In order to be a proper §102(e) rejection, the reference must describe the limitations as found in claim 1. The above cited section does not.

Application No. 09/751,762
Response dated October 2, 2006
Final Office Action dated May 30, 2006

Atty. Docket No. 2207/10121

Next, the Office Action cites column 6, lines 18-26 of Eikemeyer, which state:

... flush decode logic to determine if the thread having the stalled instruction has a previous flush condition, a dispatch flush mechanism to flush the thread having the stalled instruction from the fetch stage, the decode stage, and the dispatch stage if no other previous flush condition exists or if the previous flush condition has a lower priority than the stalled instruction so that the processor can process another of the independent threads of execution with the processor pipeline.

The cited section of Eikemeyer is intended to describe a) flush decode logic used to make the conditional determination as to whether there is a flush condition associated with a stalled instruction and b) a dispatch flush mechanism to flush the thread with the instruction.

Applicants submit that the cited sections fail to disclose the relevant limitations. The cited reference does not describe "...flushing an instruction from said first thread from a pipeline of said processing system *when data is to be loaded from [a] memory device before executing said instruction*" (e.g., as described in claim 1). Therefore, the cited section cannot support a proper §102(e) rejection of claim 1.

The Office Action further cites column 11, lines 41-49 of Eikemeyer. The cited section states:

The dispatch flush causes that thread having an instruction that is stalled at the dispatch unit 220 to be flushed from the instruction fetch and decode portions of the pipeline and all other portions of the processor's pipeline up to and including the dispatch unit 220. A flush control signal 262 causes both a dispatch flush and a normal flush/refetch 238 to flush instructions of a particular thread from the dispatch unit 220 and earlier pipeline stages.

The cited section describes flushing a thread (including an instruction) from the pipeline and the use of a flush control signal to do so. Applicants submit this section fails to support a proper §102(e) rejections for similar to those reasons detailed above. The cited section does not describe flushing an instruction *when data is to be loaded from a memory device before executing the instruction*, as specifically recited in the claim 1.

Application No. 09/751,762
Response dated October 2, 2006
Final Office Action dated May 30, 2006

Atty. Docket No. 2207/10121

The Office Action further cites column 10, lines 35-60, which state:

Some of the reasons an instruction may stall at the dispatch unit 220 include the instruction tracking queues being blocked, such as being full or otherwise unavailable to the instruction, a sync instruction being unable to execute, resources required by the instruction being blocked as in being full or otherwise unavailable, moving a thread state to/from non-renamed registers, etc. If either a dedicated LRQ 244 or dedicated SRQ 246 or dedicated GCT 248 is blocked, then the pipeline stalls behind these queues. If there are insufficient renamed pools 250 available, the pipeline stalls. Non-renamed registers may also stall the processor pipeline if some process is serialized on data or other required resource that is not renamed, e.g., Q1 must occur before Q2. In some instances, issue queues 222 may be full and/or waiting for operands to proceed and the pipeline stalls. One of the simpler instructions of a group derived from a multiple or complex instruction may stall in the dispatch stage. A sync and/or other instructions that synchronize and access storage in out-of-order processors may stall the pipeline because it needs additional information or because it must wait at dispatch stage because of downstream activity of its own thread such as clearing out a LRQ 244 or a SRQ 246. The dispatch flush aspect of the invention applies to any condition that causes a processor's pipeline to stall at dispatch when the condition causing the stall is specific to a thread.

Applicants submit the cited section is directed toward why a instruction may stall. The first portion of the cited section offers multiple examples, such as instruction tracking queues being blocked and dedicated LRQs, SRQs or GCTs being clocked. The second portion of the cited section gives further examples of conditions leading to stalls and discusses the ramifications of such a pipeline stall. The final sentence of the cited section contains a general reference to a "dispatch flush" that is applied in response to a stall when the condition causing the stall is specific to a thread. Again, Applicants submit this section fails to describe at least flushing an instruction when data *is to be loaded from a memory device before executing the instruction*, as specifically recited in the claim 1.

The Office Action further cites column 5, lines 2-6, which state:

The stalled instruction may be an instruction requiring synchronized load/storage operations and the operations are delayed. The stalled instruction may be a first instruction of a group of microcoded instructions.

Application No. 09/751,762
Response dated October 2, 2006
Final Office Action dated May 30, 2006

Atty. Docket No. 2207/10121

The cited section of Eikemeyer further describes conditions leading to a stall. It describes the stalled instruction may be an instruction requiring synchronization, and the stalled instruction may be a first instruction of a group of microcoded instructions. Applicants submit the cited section fails to describe at least flushing an instruction when *data is to be loaded from a memory device before executing the instruction*, as specifically recited in the claim 1.

Finally, the Office Action cites column 9, lines 59-65, which state:

A multithread embodiment of the invention may presume that during any given processor clock cycle an instruction from only one thread is dispatched from the dispatch unit 220 to the issue queues 222. In accordance with the dispatch flush feature of the invention, instructions of a stalled thread are flushed from the dispatch unit 220.

The final cited section describes that at any given processor cycle one instruction from only one thread is dispatched the issue queue. It further states instructions may be flushed from the dispatched unit 220 if they are stalled. Again, similar to the sections above, Applicants submit the general references to "stalls" and "flushing" in a particular context suitable to the Eikemeyer are insufficient to support a proper rejection of the embodiment of claim 1. In order to support a proper rejection, the cited reference must show at least flushing an instruction when *data is to be loaded from a memory device before executing the instruction*. Eikemeyer, for at least the reasons described above, does not.

Therefore, since each and every limitations is not found in the cited reference, the cited reference cannot adequately form the basis of a proper 35 U.S.C. §102(e) rejection of independent claim 1. Independent claims 5, 10, and 16 contain substantively similar limitations and therefore are also allowable for similar reasons. Claims 2-4, 6-9, 11-15 and 17-21 depend from allowable independent claims 1, 5, 10 and 16, and therefore are in condition for allowance as well.

Application No. 09/751,762
Response dated October 2, 2006
Final Office Action dated May 30, 2006

Atty. Docket No. 2207/10121

For at least the above reasons, Applicants respectfully submit that this application is in condition for allowance. A Notice of Allowance is earnestly solicited.

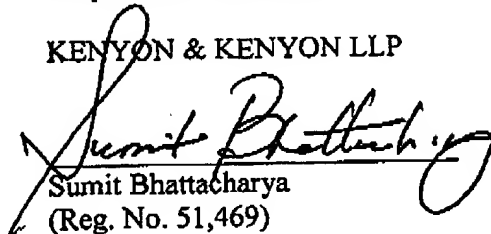
The Examiner is invited to contact the undersigned at (408) 975-7500 to discuss any matter concerning this application. The Office is hereby authorized to charge any additional fees or credit any overpayments to Deposit Account No. 11-0600.

Respectfully submitted,

KENYON & KENYON LLP

Date: October 2, 2006

By:


Sumit Bhattacharya
(Reg. No. 51,469)

KENYON & KENYON LLP
333 West San Carlos St., Suite 600
San Jose, CA 95110
Telephone: (408) 975-7500
Facsimile: (408) 975-7501

92943